

WHAT IS CLAIMED IS:

- 1 1. A method comprising:
 - 2 selecting a first allocated memory block from a plurality of allocated memory
 - 3 blocks, wherein the first allocated memory block includes a first
 - 4 allocated memory block address;
 - 5 searching other allocated memory blocks of the plurality of allocated memory
 - 6 blocks for a reference to the first allocated memory block;
 - 7 verifying that the first allocated memory block is a memory leak when the
 - 8 reference to the first allocated memory block is not found in the other
 - 9 allocated memory blocks of the plurality of allocated memory blocks;
 - 10 and
 - 11 reporting the first allocated memory block as a memory leak.
- 1 2. The method of claim 1 wherein the selecting the first allocated memory
- 2 block from the plurality of allocated memory blocks further comprises:
 - 3 selecting the first allocated memory block address from operating system
 - 4 memory management information.
- 1 3. The method of claim 1 wherein each of the plurality of allocated memory
- 2 blocks includes a header portion, and wherein the searching other allocated memory
- 3 blocks of the plurality of allocated memory blocks further comprises:
 - 4 searching the header portions of the other allocated memory blocks of the
 - 5 plurality of allocated memory blocks for a reference to the first
 - 6 allocated memory block.
- 1 4. The method of claim 1 wherein the searching other allocated memory
- 2 blocks of the plurality of allocated memory blocks further comprises:
 - 3 searching for an occurrence of the first allocated memory block address in the
 - 4 other allocated memory blocks of the plurality of allocated memory
 - 5 blocks.
- 1 5. The method of claim 1 further comprising:
 - 2 examining a reference counter corresponding to the first allocated memory
 - 3 block.

- 1 6. The method of claim 1 wherein the verifying that the first allocated
2 memory block is a memory leak further comprises:
3 determining whether the first allocated memory block has been deallocated.
- 1 7. The method of claim 1 wherein the first allocated memory block includes a
2 header portion, and wherein the verifying that the first allocated memory block is a
3 memory leak further comprises:
4 examining the header portion of the first allocated memory block.
- 1 8. The method of claim 1 wherein the verifying that the first allocated
2 memory block is a memory leak further comprises:
3 examining free block memory management information maintained by an
4 operating system.
- 1 9. The method of claim 1 wherein the reporting the first allocated memory
2 block as a memory leak further comprises:
3 displaying to a user at least one of: a program counter value, a process
4 identification value, a process name, an initial block count, a previous
5 block count, a current block count, a linearity value, the first allocated
6 memory block address, and contents of the first allocated memory
7 block.
- 1 10. The method of claim 1 wherein the reporting the first allocated memory
2 block as a memory leak further comprises:
3 storing in a data structure at least one of: a program counter value, a process
4 identification value, a process name, an initial block count, a previous
5 block count, a current block count, a linearity value, the first allocated
6 memory block address, and contents of the first allocated memory
7 block.
- 1 11. The method of claim 1 further comprising:
2 searching the first allocated memory block for a reference to at least one of the
3 plurality allocated memory blocks; and
4 storing the first allocated memory block address in a contingency chain
5 corresponding to the at least one of the plurality allocated memory

6 blocks when a reference to the at least one of the plurality allocated
7 memory blocks is found in the first allocated memory block.

1 12. The method of claim 1 further comprising:
2 examining a contingency chain corresponding to one of the plurality of
3 allocated memory blocks to determine whether any of the plurality of
4 allocated memory blocks references the one of the plurality of
5 allocated memory.

1 13. The method of claim 1 further comprising:
2 forming a contingency chain for each of the plurality of allocated memory
3 blocks, wherein each contingency chain is indexed by an allocated
4 memory block address of the corresponding each of the plurality of
5 allocated memory blocks.

1 14. A system comprising:
2 a memory;
3 a processor coupled to the memory; and
4 a memory leak detection system (MLDS) engine, wherein at least a portion of
5 the MLDS engine is encoded as instructions stored in the memory and
6 executable on the processor, and wherein the MLDS engine is
7 configured to:
8 select a first allocated memory block from a plurality of allocated
9 memory blocks stored in the memory, wherein the first
10 allocated memory block includes a first allocated memory
11 block address;
12 search other allocated memory blocks of the plurality of allocated
13 memory blocks for a reference to the first allocated memory
14 block;
15 verify that the first allocated memory block is a memory leak when the
16 reference to the first allocated memory block is not found in the
17 other allocated memory blocks of the plurality of allocated
18 memory blocks; and
19 report the first allocated memory block as a memory leak.

1 15. The system of claim 14 further comprising at least of an MLDS data
2 structure application programming interface (API), an MLDS command API, an
3 MLDS data structure, and a command line interface (CLI) parser stored in at least one
4 of the memory and a storage device accessible by the processor.

1 16. The system of claim 14 wherein the MLDS engine is further configured
2 to:

3 select the first allocated memory block address from operating system memory
4 management information.

1 17. The system of claim 14 wherein each of the plurality of allocated memory
2 blocks includes a header portion, and wherein the MLDS engine is further configured
3 to:

4 search the header portions of the other allocated memory blocks of the
5 plurality of allocated memory blocks for a reference to the first
6 allocated memory block.

1 18. The system of claim 14 wherein the MLDS engine is further configured
2 to:

3 search for an occurrence of the first allocated memory block address in the
4 other allocated memory blocks of the plurality of allocated memory
5 blocks.

1 19. The system of claim 14 wherein the MLDS engine is further configured
2 to:

3 examine a reference counter corresponding to the first allocated memory
4 block.

1 20. The system of claim 14 wherein the MLDS engine is further configured to
2 determine whether the first allocated memory block has been deallocated.

1 21. The system of claim 14 wherein the first allocated memory block includes
2 a header portion, and wherein the MLDS engine is further configured to:
3 examine the header portion of the first allocated memory block.

1 22. The system of claim 14 wherein the MLDS engine is further configured

2 to:

3 examine free block memory management information maintained by an
4 operating system.

1 23. The system of claim 14 wherein the MLDS engine is further configured

2 to:

3 display at least one of: a program counter value, a process identification value,
4 a process name, an initial block count, a previous block count, a
5 current block count, a linearity value, the first allocated memory block
6 address, and contents of the first allocated memory block.

1 24. The system of claim 14 wherein the MLDS engine is further configured

2 to:

3 store in a data structure at least one of: a program counter value, a process
4 identification value, a process name, an initial block count, a previous
5 block count, a current block count, a linearity value, the first allocated
6 memory block address, and contents of the first allocated memory
7 block.

1 25. The system of claim 14 wherein the MLDS engine is further configured

2 to:

3 search the first allocated memory block for a reference to at least one of the
4 plurality allocated memory blocks; and

5 store the first allocated memory block address in a contingency chain
6 corresponding to the at least one of the plurality allocated memory
7 blocks when a reference to the at least one of the plurality allocated
8 memory blocks is found in the first allocated memory block.

1 26. The system of claim 14 wherein the MLDS engine is further configured

2 to:

3 examine a contingency chain corresponding to one of the plurality of allocated
4 memory blocks to determine whether any of the plurality of allocated

5 memory blocks references the one of the plurality of allocated
6 memory.

1 27. The system of claim 14 wherein the MLDS engine is further configured
2 to:
3 form a contingency chain for each of the plurality of allocated memory blocks,
4 wherein each contingency chain is indexed by an allocated memory
5 block address of the corresponding each of the plurality of allocated
6 memory blocks.

1 28. A computer readable medium comprising program instructions executable
2 on a processor, the computer readable medium being at least one of an electronic
3 storage medium, a magnetic storage medium, an optical storage medium, and a
4 communications medium conveying signals encoding the instructions, wherein the
5 program instructions are operable to implement each of:
6 selecting a first allocated memory block from a plurality of allocated memory
7 blocks, wherein the first allocated memory block includes a first
8 allocated memory block address;
9 searching other allocated memory blocks of the plurality of allocated memory
10 blocks for a reference to the first allocated memory block;
11 verifying that the first allocated memory block is a memory leak when the
12 reference to the first allocated memory block is not found in the other
13 allocated memory blocks of the plurality of allocated memory blocks;
14 and
15 reporting the first allocated memory block as a memory leak.

1 29. The computer readable medium of claim 28 wherein the selecting the first
2 allocated memory block from the plurality of allocated memory blocks further
3 comprises:
4 selecting the first allocated memory block address from operating system
5 memory management information.

1 30. The computer readable medium of claim 28 wherein each of the plurality
2 of allocated memory blocks includes a header portion, and wherein the searching

3 other allocated memory blocks of the plurality of allocated memory blocks further
4 comprises:

5 searching the header portions of the other allocated memory blocks of the
6 plurality of allocated memory blocks for a reference to the first
7 allocated memory block.

1 31. The computer readable medium of claim 28 wherein the searching other
2 allocated memory blocks of the plurality of allocated memory blocks further
3 comprises:

4 searching for an occurrence of the first allocated memory block address in the
5 other allocated memory blocks of the plurality of allocated memory
6 blocks.

1 32. The computer readable medium of claim 28 further comprising program
2 instructions are operable to implement:

3 examining a reference counter corresponding to the first allocated memory
4 block.

1 33. The computer readable medium of claim 28 wherein the verifying that the
2 first allocated memory block is a memory leak further comprises:

3 determining whether the first allocated memory block has been deallocated.

1 34. The computer readable medium of claim 28 wherein the first allocated
2 memory block includes a header portion, and wherein the verifying that the first
3 allocated memory block is a memory leak further comprises:

4 examining the header portion of the first allocated memory block.

1 35. The computer readable medium of claim 28 wherein the verifying that the
2 first allocated memory block is a memory leak further comprises:

3 examining free block memory management information maintained by an
4 operating system.

1 36. The computer readable medium of claim 28 wherein the reporting the first
2 allocated memory block as a memory leak further comprises:

3 displaying to a user at least one of: a program counter value, a process
4 identification value, a process name, an initial block count, a previous

5 block count, a current block count, a linearity value, the first allocated
6 memory block address, and contents of the first allocated memory
7 block.

1 37. The computer readable medium of claim 28 wherein the reporting the first
2 allocated memory block as a memory leak further comprises:

3 storing in a data structure at least one of: a program counter value, a process
4 identification value, a process name, an initial block count, a previous
5 block count, a current block count, a linearity value, the first allocated
6 memory block address, and contents of the first allocated memory
7 block.

1 38. The computer readable medium of claim 28 further comprising program
2 instructions are operable to implement each of:

3 searching the first allocated memory block for a reference to at least one of the
4 plurality allocated memory blocks; and
5 storing the first allocated memory block address in a contingency chain
6 corresponding to the at least one of the plurality allocated memory
7 blocks when a reference to the at least one of the plurality allocated
8 memory blocks is found in the first allocated memory block.

1 39. The computer readable medium of claim 28 further comprising program
2 instructions are operable to implement each of:

3 examining a contingency chain corresponding to one of the plurality of
4 allocated memory blocks to determine whether any of the plurality of
5 allocated memory blocks references the one of the plurality of
6 allocated memory.

1 40. The computer readable medium of claim 28 further comprising program
2 instructions are operable to implement:

3 forming a contingency chain for each of the plurality of allocated memory
4 blocks, wherein each contingency chain is indexed by an allocated
5 memory block address of the corresponding each of the plurality of
6 allocated memory blocks.

- 1 41. An apparatus comprising:
- 2 a means for selecting a first allocated memory block from a plurality of
- 3 allocated memory blocks, wherein the first allocated memory block
- 4 includes a first allocated memory block address;
- 5 a means for searching other allocated memory blocks of the plurality of
- 6 allocated memory blocks for a reference to the first allocated memory
- 7 block;
- 8 a means for verifying that the first allocated memory block is a memory leak
- 9 when the reference to the first allocated memory block is not found in
- 10 the other allocated memory blocks of the plurality of allocated memory
- 11 blocks; and
- 12 a means for reporting the first allocated memory block as a memory leak.
- 1 42. The apparatus of claim 41 wherein each of the plurality of allocated
- 2 memory blocks includes a header portion, and wherein the apparatus further
- 3 comprises:
- 4 a means for searching the header portions of the other allocated memory
- 5 blocks of the plurality of allocated memory blocks for a reference to
- 6 the first allocated memory block.
- 1 43. The apparatus of claim 41 further comprising:
- 2 a means for searching for an occurrence of the first allocated memory block
- 3 address in the other allocated memory blocks of the plurality of
- 4 allocated memory blocks.
- 1 44. The apparatus of claim 41 further comprising:
- 2 a means for displaying to a user at least one of: a program counter value, a
- 3 process identification value, a process name, an initial block count, a
- 4 previous block count, a current block count, a linearity value, the first
- 5 allocated memory block address, and contents of the first allocated
- 6 memory block.

1 45. The apparatus of claim 41 further comprising:
2 a means for searching the first allocated memory block for a reference to at
3 least one of the plurality allocated memory blocks; and
4 a means for storing the first allocated memory block address in a contingency
5 chain corresponding to the at least one of the plurality allocated
6 memory blocks when a reference to the at least one of the plurality
7 allocated memory blocks is found in the first allocated memory block.

1 46. The apparatus of claim 41 further comprising:
2 a means for examining a contingency chain corresponding to one of the
3 plurality of allocated memory blocks to determine whether any of the
4 plurality of allocated memory blocks references the one of the plurality
5 of allocated memory.

1 47. The apparatus of claim 41 further comprising:
2 a means for forming a contingency chain for each of the plurality of allocated
3 memory blocks, wherein each contingency chain is indexed by an
4 allocated memory block address of the corresponding each of the
5 plurality of allocated memory blocks.